Week 9 Exercises: Propositional Logic - Detailed Solutions

> **Note:** This document provides comprehensive step-by-step solutions for all exercises in Week 9. These solutions are intended for instructor reference and can be shared with students as needed.

Part I: Translation from English to Logic

Translate each sentence into propositional logic using noun-based propositional atoms taken directly from the sentence.

1. If it rains, the sky is cloudy.
2. If An had a lot of money, she would go shopping.
3. Monkeys are not literate.
4. The server is online and the database is accessible.
5. Either the model overfits or the dataset is too small.
6. If the lecture starts late, the lab session will be shorter.
7. The patient has a fever if and only if the thermometer reads above 38°C.
8. It is not the case that both the code compiles and the tests fail.
9. If the assignment is submitted, then the checker runs, and if it runs, a report is generated.
10. Either the network is down or the credentials are incorrect, but not both.

Part II: Convert to CNF (revised, higher difficulty)

For each formula, convert to CNF showing standard steps in solutions: eliminate $\Leftrightarrow$, $\Rightarrow$, push negations inward, remove double negations, and distribute $\vee$ over $\wedge$ until CNF is reached.

11. $(P \Rightarrow (Q \wedge R)) \vee (S \Rightarrow Q)$.
12. $((P \vee Q) \Rightarrow (R \wedge S))$.
13. $(P \Leftrightarrow (Q \vee R)) \wedge (\neg R \vee S)$.
14. $\neg(P \Rightarrow (Q \Rightarrow R)) \vee S$.
15. $(P \Leftrightarrow Q) \Rightarrow (R \vee S)$.
16. $(P \Rightarrow Q) \wedge (Q \Leftrightarrow (R \wedge S))$.
17. $(P \Rightarrow (Q \Leftrightarrow R)) \vee (S \wedge \neg R)$.
18. $\neg((P \vee Q) \Leftrightarrow R)$.
19. $(P \vee (Q \Rightarrow R)) \Rightarrow (S \vee R)$.
20. $((P \wedge \neg Q) \Rightarrow (R \vee S)) \wedge (Q \vee R)$.

Part III: Inference rule proofs

Use rules such as Modus Ponens, And-Elimination/Introduction, and Resolution to derive the target conclusion from the premises.
21. Premises: P ⇒ Q; Q ⇒ R; P — Prove: R.
22. Premises: P ∧ Q; P ⇒ R; Q ⇒ S — Prove: R ∧ S.
23. Premises: P ⇔ Q; P — Prove: Q.
24. Premises: P ∨ Q; ¬P ∨ R; ¬Q — Prove: R.
25. Premises: P ⇒ (Q ∨ R); ¬Q; R ⇒ S; P — Prove: S.

Part IV: Translation + CNF (combined)

For each item, first translate the English sentence into logic using noun-based atoms, then convert that translation to CNF using the standard procedure; show both steps in your solution write-up.
26. If the sensor detects smoke, then the alarm sounds and the sprinklers activate.
27. The system is online if and only if the database is reachable or the cache is warm.
28. It is not the case that both the model converges and the validation error increases.
29. If the user is authenticated and not banned, then the action is permitted or audited.
30. Either the API returns 200 or, if it returns 500, then alerts trigger.
31. If the build passes, then deployment proceeds if and only if approvals are granted.
32. Unless the battery is charged, the device powers off.
33. The dataset is large if and only if the sample size is above 10k and the features exceed 100.
34. The lab is open implies the TA is present or the sign shows remote support, and the TA being present implies the lab is open.
35. It is not the case that if the server is down or the firewall blocks traffic, then the website is reachable.

Detailed Solutions

Part I: Translation from English to Logic

Exercise 1: If it rains, the sky is cloudy.

- **Solution:** This is a conditional statement (if-then).
- **Atoms:** Rains = "It rains", SkyCloudy = "The sky is cloudy"
- **Answer:** Rains ⇒ SkyCloudy

Exercise 2: If An had a lot of money, she would go shopping.

- **Solution:** Conditional statement with proper nouns.

- **Atoms:** AnHasALotOfMoney = "An had a lot of money", AnGoesShopping = "An would go shopping"
- **Answer:** AnHasALotOfMoney ⇒ AnGoesShopping

Exercise 3: Monkeys are not literate.

- **Solution:** Negation of a statement.
- **Atoms:** MonkeysAreLiterate = "Monkeys are literate"
- **Answer:** ¬MonkeysAreLiterate

Exercise 4: The server is online and the database is accessible.

- **Solution:** Conjunction (and) of two statements.
- **Atoms:** ServerOnline = "The server is online", DatabaseAccessible = "The database is accessible"
- **Answer:** ServerOnline ∧ DatabaseAccessible

Exercise 5: Either the model overfits or the dataset is too small.

- **Solution:** Disjunction (or) of two statements.
- **Atoms:** ModelOverfits = "The model overfits", DatasetTooSmall = "The dataset is too small"
- **Answer:** ModelOverfits ∨ DatasetTooSmall

Exercise 6: If the lecture starts late, the lab session will be shorter.

- **Solution:** Conditional statement.
- **Atoms:** LectureStartsLate = "The lecture starts late", LabSessionShorter = "The lab session will be shorter"
- **Answer:** LectureStartsLate ⇒ LabSessionShorter

Exercise 7: The patient has a fever if and only if the thermometer reads above 38°C.

- **Solution:** Biconditional (if and only if).
- **Atoms:** PatientHasFever = "The patient has a fever", ThermometerAbove38C = "The thermometer reads above 38°C"
- **Answer:** PatientHasFever ⇔ ThermometerAbove38C

Exercise 8: It is not the case that both the code compiles and the tests fail.

- **Solution:** Negation of a conjunction.
- **Atoms:** CodeCompiles = "The code compiles", TestsFail = "The tests fail"
- **Answer:** ¬(CodeCompiles ∧ TestsFail)

Exercise 9: If the assignment is submitted, then the checker runs, and if it runs, a report is generated.

- **Solution:** Conjunction of two conditional statements.

- **Atoms:** AssignmentSubmitted = "The assignment is submitted", CheckerRuns = "The checker runs", ReportGenerated = "A report is generated"
- **Answer:** (AssignmentSubmitted ⇒ CheckerRuns) ∧ (CheckerRuns ⇒ ReportGenerated)

Exercise 10: Either the network is down or the credentials are incorrect, but not both.

- **Solution:** Exclusive or (XOR) - can be expressed as (A ∨ B) ∧ ¬(A ∧ B).
- **Atoms:** NetworkDown = "The network is down", CredentialsIncorrect = "The credentials are incorrect"
- **Answer:** (NetworkDown ∨ CredentialsIncorrect) ∧ ¬(NetworkDown ∧ CredentialsIncorrect)

Part II: Convert to CNF

Exercise 11: (P ⇒ (Q ∧ R)) ∨ (S ⇒ Q)

Step-by-step solution:

1. **Eliminate implications:** Replace α ⇒ β with ¬α ∨ β
    - P ⇒ (Q ∧ R) becomes ¬P ∨ (Q ∧ R)
    - S ⇒ Q becomes ¬S ∨ Q
    - Result: (¬P ∨ (Q ∧ R)) ∨ (¬S ∨ Q)
2. **Distribute ∨ over ∧:** Apply α ∨ (β ∧ γ) = (α ∨ β) ∧ (α ∨ γ)
    - (¬P ∨ (Q ∧ R)) ∨ (¬S ∨ Q) = (¬P ∨ Q ∧ R ∨ ¬S ∨ Q)
    - Distribute: (¬P ∨ Q ∨ ¬S ∨ Q) ∧ (¬P ∨ R ∨ ¬S ∨ Q)
3. **Simplify:** Remove duplicate literals
    - (¬P ∨ Q ∨ ¬S ∨ Q) = (¬P ∨ Q ∨ ¬S)
    - (¬P ∨ R ∨ ¬S ∨ Q) = (¬P ∨ Q ∨ R ∨ ¬S)

Final CNF: (¬P ∨ Q ∨ ¬S) ∧ (¬P ∨ Q ∨ R ∨ ¬S)

Exercise 12: ((P ∨ Q) ⇒ (R ∧ S))

Step-by-step solution:

1. **Eliminate implication:** ¬(P ∨ Q) ∨ (R ∧ S)
2. **Move negation inward (De Morgan's):** ¬(P ∨ Q) = ¬P ∧ ¬Q
    - Result: (¬P ∧ ¬Q) ∨ (R ∧ S)

3. **Distribute ∨ over ∧:**
    - (¬P ∧ ¬Q) ∨ (R ∧ S) = (¬P ∨ R ∧ S) ∧ (¬Q ∨ R ∧ S)
    - Further distribute: (¬P ∨ R) ∧ (¬P ∨ S) ∧ (¬Q ∨ R) ∧ (¬Q ∨ S)

Final CNF: (¬P ∨ R) ∧ (¬P ∨ S) ∧ (¬Q ∨ R) ∧ (¬Q ∨ S)

Exercise 13: (P ⇔ (Q ∨ R)) ∧ (¬R ∨ S)

Step-by-step solution:

1. **Eliminate biconditional:** P ⇔ (Q ∨ R) becomes (P ⇒ (Q ∨ R)) ∧ ((Q ∨ R) ⇒ P)
    - Result: ((P ⇒ (Q ∨ R)) ∧ ((Q ∨ R) ⇒ P)) ∧ (¬R ∨ S)
2. **Eliminate implications:**
    - P ⇒ (Q ∨ R) becomes ¬P ∨ Q ∨ R
    - (Q ∨ R) ⇒ P becomes ¬(Q ∨ R) ∨ P = (¬Q ∧ ¬R) ∨ P
    - Result: (¬P ∨ Q ∨ R) ∧ ((¬Q ∧ ¬R) ∨ P) ∧ (¬R ∨ S)
3. **Distribute ∨ over ∧:**
    - ((¬Q ∧ ¬R) ∨ P) = (¬Q ∨ P) ∧ (¬R ∨ P)
    - Result: (¬P ∨ Q ∨ R) ∧ (¬Q ∨ P) ∧ (¬R ∨ P) ∧ (¬R ∨ S)

Final CNF: (¬P ∨ Q ∨ R) ∧ (¬Q ∨ P) ∧ (¬R ∨ P) ∧ (¬R ∨ S)

Exercise 14: ¬(P ⇒ (Q ⇒ R)) ∨ S

Step-by-step solution:

1. **Eliminate inner implication:** Q ⇒ R becomes ¬Q ∨ R
    - Result: ¬(P ⇒ (¬Q ∨ R)) ∨ S
2. **Eliminate outer implication:** P ⇒ (¬Q ∨ R) becomes ¬P ∨ ¬Q ∨ R
    - Result: ¬(¬P ∨ ¬Q ∨ R) ∨ S
3. **Move negation inward (De Morgan's):** ¬(¬P ∨ ¬Q ∨ R) = P ∧ Q ∧ ¬R
    - Result: (P ∧ Q ∧ ¬R) ∨ S
4. **Distribute ∨ over ∧:**
    - (P ∧ Q ∧ ¬R) ∨ S = (P ∨ S) ∧ (Q ∨ S) ∧ (¬R ∨ S)

Final CNF: (P ∨ S) ∧ (Q ∨ S) ∧ (¬R ∨ S)

Exercise 15: $(P \Leftrightarrow Q) \Rightarrow (R \lor S)$

Step-by-step solution:

1. **Eliminate biconditional:** $P \Leftrightarrow Q$ becomes $(P \Rightarrow Q) \land (Q \Rightarrow P)$
    - Result: $((P \Rightarrow Q) \land (Q \Rightarrow P)) \Rightarrow (R \lor S)$
2. **Eliminate outer implication:** $\neg((P \Rightarrow Q) \land (Q \Rightarrow P)) \lor (R \lor S)$
    - Eliminate inner implications: $(\neg P \lor Q) \land (\neg Q \lor P)$
    - Result: $\neg((\neg P \lor Q) \land (\neg Q \lor P)) \lor (R \lor S)$
3. **Move negation inward (De Morgan's):** $\neg((\neg P \lor Q) \land (\neg Q \lor P)) = \neg(\neg P \lor Q) \lor \neg(\neg Q \lor P)$
    - $\neg(\neg P \lor Q) = P \land \neg Q$
    - $\neg(\neg Q \lor P) = Q \land \neg P$
    - Result: $(P \land \neg Q) \lor (Q \land \neg P) \lor (R \lor S)$
4. **Distribute $\lor$ over $\land$:**
    - $(P \land \neg Q) \lor (Q \land \neg P) \lor (R \lor S) = (P \lor Q \lor R \lor S) \land (\neg Q \lor Q \lor R \lor S) \land (P \lor \neg P \lor R \lor S) \land (\neg Q \lor \neg P \lor R \lor S)$
    - Simplify: $(P \lor Q \lor R \lor S) \land (R \lor S) \land (R \lor S) \land (\neg Q \lor \neg P \lor R \lor S)$
    - Further simplify: $(P \lor Q \lor R \lor S) \land (R \lor S) \land (\neg P \lor \neg Q \lor R \lor S)$

Final CNF: $(P \lor Q \lor R \lor S) \land (R \lor S) \land (\neg P \lor \neg Q \lor R \lor S)$

Part III: Inference Rule Proofs

Exercise 21: Premises: $P \Rightarrow Q$; $Q \Rightarrow R$; P — Prove: R

Solution using Modus Ponens:

1. $P \Rightarrow Q$ (Premise 1)
2. $Q \Rightarrow R$ (Premise 2)
3. P (Premise 3)
4. Q (From 1,3 by Modus Ponens)
5. R (From 2,4 by Modus Ponens)

Answer: R

Exercise 22: Premises: $P \land Q$; $P \Rightarrow R$; $Q \Rightarrow S$ — Prove: $R \land S$

Solution:

1. $P \land Q$ (Premise 1)

2. P ⇒ R (Premise 2)
3. Q ⇒ S (Premise 3)
4. P (From 1 by And-Elimination)
5. Q (From 1 by And-Elimination)
6. R (From 2,4 by Modus Ponens)
7. S (From 3,5 by Modus Ponens)
8. R ∧ S (From 6,7 by And-Introduction)

Answer: R ∧ S

Exercise 23: Premises: P ⇔ Q; P — Prove: Q

Solution:

1. P ⇔ Q (Premise 1)
2. P (Premise 2)
3. P ⇒ Q (From 1 by Biconditional-Elimination)
4. Q (From 2,3 by Modus Ponens)

Answer: Q

Exercise 24: Premises: P ∨ Q; ¬P ∨ R; ¬Q — Prove: R

Solution using Resolution:

1. P ∨ Q (Premise 1)
2. ¬P ∨ R (Premise 2)
3. ¬Q (Premise 3)
4. Q ∨ R (From 1,2 by Resolution on P)
5. R (From 3,4 by Resolution on Q)

Answer: R

Exercise 25: Premises: P ⇒ (Q ∨ R); ¬Q; R ⇒ S; P — Prove: S

Solution:

1. P ⇒ (Q ∨ R) (Premise 1)
2. ¬Q (Premise 2)
3. R ⇒ S (Premise 3)
4. P (Premise 4)
5. Q ∨ R (From 1,4 by Modus Ponens)
6. R (From 2,5 by Disjunctive Syllogism)
7. S (From 3,6 by Modus Ponens)

Answer: S

Part IV: Translation + CNF (Combined)

Exercise 26: If the sensor detects smoke, then the alarm sounds and the sprinklers activate.

Translation:

- **Atoms:** SensorDetectsSmoke = "The sensor detects smoke", AlarmSounds = "The alarm sounds", SprinklersActivate = "The sprinklers activate"
- **Logical form:** SensorDetectsSmoke ⇒ (AlarmSounds ∧ SprinklersActivate)

CNF Conversion:

1. Eliminate implication: ¬SensorDetectsSmoke ∨ (AlarmSounds ∧ SprinklersActivate)
2. Distribute ∨ over ∧: (¬SensorDetectsSmoke ∨ AlarmSounds) ∧ (¬SensorDetectsSmoke ∨ SprinklersActivate)

Final Answer: Translation: SensorDetectsSmoke ⇒ (AlarmSounds ∧ SprinklersActivate); CNF: (¬SensorDetectsSmoke ∨ AlarmSounds) ∧ (¬SensorDetectsSmoke ∨ SprinklersActivate)

Exercise 27: The system is online if and only if the database is reachable or the cache is warm.

Translation:

- **Atoms:** SystemOnline = "The system is online", DatabaseReachable = "The database is reachable", CacheWarm = "The cache is warm"
- **Logical form:** SystemOnline ⇔ (DatabaseReachable ∨ CacheWarm)

CNF Conversion:

1. Eliminate biconditional: (SystemOnline ⇒ (DatabaseReachable ∨ CacheWarm)) ∧ ((DatabaseReachable ∨ CacheWarm) ⇒ SystemOnline)
2. Eliminate implications: (¬SystemOnline ∨ DatabaseReachable ∨ CacheWarm) ∧ (¬(DatabaseReachable ∨ CacheWarm) ∨ SystemOnline)
3. Move negation inward: (¬SystemOnline ∨ DatabaseReachable ∨ CacheWarm) ∧ ((¬DatabaseReachable ∧ ¬CacheWarm) ∨ SystemOnline)
4. Distribute ∨ over ∧: (¬SystemOnline ∨ DatabaseReachable ∨ CacheWarm) ∧ (¬DatabaseReachable ∨ SystemOnline) ∧ (¬CacheWarm ∨ SystemOnline)

Final Answer: Translation: SystemOnline ⇔ (DatabaseReachable ∨ CacheWarm); CNF: (¬SystemOnline ∨ DatabaseReachable ∨ CacheWarm) ∧ (¬DatabaseReachable ∨ SystemOnline) ∧ (¬CacheWarm ∨ SystemOnline)

Exercise 28: It is not the case that both the model converges and the validation error increases.

Translation:

- **Atoms:** ModelConverges = "The model converges", ValidationErrorIncreases = "The validation error increases"
- **Logical form:** ¬(ModelConverges ∧ ValidationErrorIncreases)

CNF Conversion:

1. Apply De Morgan's law: ¬ModelConverges ∨ ¬ValidationErrorIncreases

Final Answer: Translation: ¬(ModelConverges ∧ ValidationErrorIncreases); CNF: (¬ModelConverges ∨ ¬ValidationErrorIncreases)

Exercise 29: If the user is authenticated and not banned, then the action is permitted or audited.

Translation:

- **Atoms:** UserAuthenticated = "The user is authenticated", UserBanned = "The user is banned", ActionPermitted = "The action is permitted", ActionAudited = "The action is audited"
- **Logical form:** (UserAuthenticated ∧ ¬UserBanned) ⇒ (ActionPermitted ∨ ActionAudited)

CNF Conversion:

1. Eliminate implication: ¬(UserAuthenticated ∧ ¬UserBanned) ∨ (ActionPermitted ∨ ActionAudited)
2. Apply De Morgan's law: (¬UserAuthenticated ∨ ¬¬UserBanned) ∨ (ActionPermitted ∨ ActionAudited)
3. Remove double negation: (¬UserAuthenticated ∨ UserBanned) ∨ ActionPermitted ∨ ActionAudited
4. Rearrange: ¬UserAuthenticated ∨ UserBanned ∨ ActionPermitted ∨ ActionAudited

Final Answer: Translation: (UserAuthenticated ∧ ¬UserBanned) ⇒ (ActionPermitted ∨ ActionAudited); CNF: (¬UserAuthenticated ∨ UserBanned ∨ ActionPermitted ∨ ActionAudited)

Exercise 30: Either the API returns 200 or, if it returns 500, then alerts trigger.

Translation:

- **Atoms:** APIReturns200 = "The API returns 200", APIReturns500 = "The API returns 500", AlertsTrigger = "Alerts trigger"
- **Logical form:** APIReturns200 ∨ (APIReturns500 ⇒ AlertsTrigger)

CNF Conversion:

1. Eliminate implication: APIReturns200 ∨ (¬APIReturns500 ∨ AlertsTrigger)
2. Rearrange: APIReturns200 ∨ ¬APIReturns500 ∨ AlertsTrigger

Final Answer: Translation: APIReturns200 ∨ (APIReturns500 ⇒ AlertsTrigger); CNF: APIReturns200 ∨ ¬APIReturns500 ∨ AlertsTrigger

Exercise 31: If the build passes, then deployment proceeds if and only if approvals are granted.

Translation:

- **Atoms:** BuildPasses = "The build passes", DeploymentProceeds = "Deployment proceeds", ApprovalsGranted = "Approvals are granted"
- **Logical form:** BuildPasses ⇒ (DeploymentProceeds ⇔ ApprovalsGranted)

CNF Conversion:

1. Eliminate biconditional: BuildPasses ⇒ ((DeploymentProceeds ⇒ ApprovalsGranted) ∧ (ApprovalsGranted ⇒ DeploymentProceeds))
2. Eliminate implications: ¬BuildPasses ∨ ((¬DeploymentProceeds ∨ ApprovalsGranted) ∧ (¬ApprovalsGranted ∨ DeploymentProceeds))
3. Distribute ∨ over ∧: (¬BuildPasses ∨ ¬DeploymentProceeds ∨ ApprovalsGranted) ∧ (¬BuildPasses ∨ ¬ApprovalsGranted ∨ DeploymentProceeds)

Final Answer: Translation: BuildPasses ⇒ (DeploymentProceeds ⇔ ApprovalsGranted); CNF: (¬BuildPasses ∨ ¬DeploymentProceeds ∨ ApprovalsGranted) ∧ (¬BuildPasses ∨ ¬ApprovalsGranted ∨ DeploymentProceeds)

Exercise 32: Unless the battery is charged, the device powers off.

Translation:

- **Note:** "Unless A, B" means "If not A, then B" or equivalently "A ∨ B"
- **Atoms:** BatteryCharged = "The battery is charged", DevicePowersOff = "The device powers off"
- **Logical form:** ¬BatteryCharged ⇒ DevicePowersOff (or equivalently BatteryCharged ∨ DevicePowersOff)

CNF Conversion:

1. Using equivalence: BatteryCharged ∨ DevicePowersOff

Final Answer: Translation: ¬BatteryCharged ⇒ DevicePowersOff (equivalently BatteryCharged ∨ DevicePowersOff); CNF: BatteryCharged ∨ DevicePowersOff

Exercise 33: The dataset is large if and only if the sample size is above 10k and the features exceed 100.

Translation:

- **Atoms:** DatasetLarge = "The dataset is large", SampleAbove10k = "The sample size is above 10k", FeaturesExceed100 = "The features exceed 100"
- **Logical form:** DatasetLarge ⇔ (SampleAbove10k ∧ FeaturesExceed100)

CNF Conversion:

1. Eliminate biconditional: (DatasetLarge ⇒ (SampleAbove10k ∧ FeaturesExceed100)) ∧ ((SampleAbove10k ∧ FeaturesExceed100) ⇒ DatasetLarge)
2. Eliminate implications: (¬DatasetLarge ∨ (SampleAbove10k ∧ FeaturesExceed100)) ∧ (¬(SampleAbove10k ∧ FeaturesExceed100) ∨ DatasetLarge)
3. Apply De Morgan's law: (¬DatasetLarge ∨ SampleAbove10k ∧ FeaturesExceed100) ∧ ((¬SampleAbove10k ∨ ¬FeaturesExceed100) ∨ DatasetLarge)
4. Distribute ∨ over ∧: (¬DatasetLarge ∨ SampleAbove10k) ∧ (¬DatasetLarge ∨ FeaturesExceed100) ∧ (¬SampleAbove10k ∨ ¬FeaturesExceed100 ∨ DatasetLarge)

Final Answer: Translation: DatasetLarge ⇔ (SampleAbove10k ∧ FeaturesExceed100); CNF: (¬DatasetLarge ∨ SampleAbove10k) ∧ (¬DatasetLarge ∨ FeaturesExceed100) ∧ (¬SampleAbove10k ∨ ¬FeaturesExceed100 ∨ DatasetLarge)

Exercise 34: The lab is open implies the TA is present or the sign shows remote support, and the TA being present implies the lab is open.

Translation:

- **Atoms:** LabOpen = "The lab is open", TAPresent = "The TA is present", SignShowsRemoteSupport = "The sign shows remote support"
- **Logical form:** (LabOpen ⇒ (TAPresent ∨ SignShowsRemoteSupport)) ∧ (TAPresent ⇒ LabOpen)

CNF Conversion:

1. Eliminate implications: (¬LabOpen ∨ TAPresent ∨ SignShowsRemoteSupport) ∧ (¬TAPresent ∨ LabOpen)

Final Answer: Translation: (LabOpen ⇒ (TAPresent ∨ SignShowsRemoteSupport)) ∧ (TAPresent ⇒ LabOpen); CNF: (¬LabOpen ∨ TAPresent ∨ SignShowsRemoteSupport) ∧ (¬TAPresent ∨ LabOpen)

Exercise 35: It is not the case that if the server is down or the firewall blocks traffic, then the website is reachable.

Translation:

- **Atoms:** ServerDown = "The server is down", FirewallBlocksTraffic = "The firewall blocks traffic", WebsiteReachable = "The website is reachable"

- **Logical form:** ¬((ServerDown ∨ FirewallBlocksTraffic) ⇒ WebsiteReachable)

CNF Conversion:

1. Eliminate implication: ¬(¬(ServerDown ∨ FirewallBlocksTraffic) ∨ WebsiteReachable)
2. Move negation inward: ¬¬(ServerDown ∨ FirewallBlocksTraffic) ∧ ¬WebsiteReachable
3. Remove double negation: (ServerDown ∨ FirewallBlocksTraffic) ∧ ¬WebsiteReachable

Final Answer: Translation: ¬((ServerDown ∨ FirewallBlocksTraffic) ⇒ WebsiteReachable); CNF: (ServerDown ∨ FirewallBlocksTraffic) ∧ (¬WebsiteReachable)